index
**app** /home/uwu/Github_Repos/University/M.O.S.I.S/M.O.S.I.S_Host_Software/App/app.py

Launches app.py using python3 without calling interpreter explicitly.

**Modules**

os

**Classes**

builtins.object

MediaEntryInternalRepresentation
MediaMetadataInternalRepresentation

enum.Enum(builtins.object)

illuminationType
shotType

sqlalchemy.orm.decl_api.Model(flask_sqlalchemy.model.Model)

MediaEntry
MediaMetadata

class **MediaEntry**(sqlalchemy.orm.decl_api.Model)
MediaEntry(**kwargs)

Create media_entry database table.

Contains:
entryId (PRIMARY KEY INTEGER),
shotType (TEXT NOT NULLABLE) an enum,
time (TEXT NOT NULLABLE) format (yyyy-MM-ddTHH:mm:ss.zzz),
illumination_type (TEXT NOT NULLABLE) an enum,
iso (INTEGER NOT NULLABLE),
apertureSize (REAL NOT NULLABLE),
shutterSpeed (REAL NOT NULLABLE),
whiteBalance (INTEGER NOT NULLABLE)

Method resolution order:
MediaEntry
sqlalchemy.orm.decl_api.Model
flask_sqlalchemy.model.Model
builtins.object

---

Methods defined here:

**__init__**(self, **kwargs)
A simple constructor that allows initialization from kwargs.

Sets attributes on the constructed instance using the names and
values in ``kwargs``.

Only keys that are present as
attributes of the instance's class are allowed. These could be,
for example, any mapped columns or relationships.

**__repr__**(self)
Return MediaEntry.entryId when inserting into database.

---

Data descriptors defined here:

**apertureSize**

**entryId**

**illuminationType**

**iso**

**shotType**

**shutterSpeed**

**time**

**whiteBalance**

---

Data and other attributes defined here:

**__mapper__** = <Mapper at 0x7f2b6551e510; MediaEntry>

**__table__** = Table('MediaEntry', MetaData(), Column('entryId'...table=<MediaEntry>, nullable=False), schema=None)

**__tablename__** = 'MediaEntry'

---

Methods inherited from sqlalchemy.orm.decl_api.Model:

**query**

---

Data and other attributes inherited from sqlalchemy.orm.decl_api.Model:

**__abstract__** = True

**__fsa__** = <SQLAlchemy>

**metadata** = MetaData()

**query_class** = <class 'flask_sqlalchemy.query.Query'>
SQLAlchemy :class:`~sqlalchemy.orm.query.Query` subclass with some extra methods
useful for querying in a web application.

This is the default query class for :attr:`.Model.query`.

.. versionchanged:: 3.0
Renamed to ``Query`` from ``BaseQuery``.

**registry** = <sqlalchemy.orm.decl_api.registry object>

---

Data descriptors inherited from flask_sqlalchemy.model.Model:

**__dict__**
dictionary for instance variables (if defined)

**__weakref__**
list of weak references to the object (if defined)

---

Data and other attributes inherited from flask_sqlalchemy.model.Model:

**__annotations__** = {'__fsa__': 't.ClassVar[SQLAlchemy]', 'query': 't.ClassVar[Query]', 'query_class': 't.ClassVar[type[Query]]'}

class **MediaEntryInternalRepresentation**(builtins.object)
MediaEntryInternalRepresentation(entryId: int, shotType: None, time: str, illuminationType: None, iso: int, apertureSize: float, shutterSpeed: float, whiteBalance: i

Internal representation for MediaEntry.

Methods defined here:

**__init__**(self, entryId: int, shotType: None, time: str, illuminationType: None, iso: int, apertureSize: float, shutterSpeed: float, whiteBalance: int)
Construct MediaEntryInternalRepesentation.

---

Data descriptors defined here:

**__dict__**
dictionary for instance variables (if defined)

**__weakref__**
list of weak references to the object (if defined)

---

Data and other attributes defined here:

**apertureSize** = 0.0

**entryId** = 0

**illuminationType** = None

**iso** = 0

**shotType** = None

**shutterSpeed** = 0.0

**time** = ''

**whiteBalance** = 0.0

class **MediaMetadata**(sqlalchemy.orm.decl_api.Model)

MediaMetadata(**kwargs)

Create MediaMetadata database table.

Contains:
metadataId (PRIMARY KEY INTEGER),
entryId (INTEGER, FOREIGN KEY (MediaEntry.entryId)),
leftCameraMedia (TEXT NOT NULLABLE),
rightCameraMedia (TEXT NOT NULLABLE),
time (TEXT NOT NULLABLE) format (yyyy-MM-ddTHH:mm:ss.zzzzzz),
temperature (REAL NOT NULLABLE),
pressure (REAL NOT NULLABLE),
ph (REAL NOT NULLABLE),
dissolvedOxygen (REAL NOT NULLABLE),

Method resolution order:
MediaMetadata
sqlalchemy.orm.decl_api.Model
flask_sqlalchemy.model.Model
builtins.object

---

Methods defined here:

**__init__**(self, **kwargs)
A simple constructor that allows initialization from kwargs.

Sets attributes on the constructed instance using the names and
values in ``kwargs``.

Only keys that are present as
attributes of the instance's class are allowed. These could be,
for example, any mapped columns or relationships.

**__repr__**(self)
Return MediaMetadata.metadataId when inserting into database.

---

Data descriptors defined here:

**dissolvedOxygen**

**entryId**

**leftCameraMedia**

**metadataId**

**ph**

**pressure**

**rightCameraMedia**

**temperature**

**time**

---

Data and other attributes defined here:

**__mapper__** = <Mapper at 0x7f2b63cba7d0; MediaMetadata>

**__table__** = Table('MediaMetadata', MetaData(), Column('metad...le=<MediaMetadata>, nullable=False), schema=None)

**__tablename__** = 'MediaMetadata'

---

Methods inherited from sqlalchemy.orm.decl_api.Model:

**query**

---

Data and other attributes inherited from sqlalchemy.orm.decl_api.Model:

**__abstract__** = True

**__fsa__** = <SQLAlchemy>

**metadata** = MetaData()

**query_class** = <class 'flask_sqlalchemy.query.Query'>
SQLAlchemy :class:`~sqlalchemy.orm.query.Query` subclass with some extra methods
useful for querying in a web application.

This is the default query class for :attr:`.Model.query`.

.. versionchanged:: 3.0
Renamed to ``Query`` from ``BaseQuery``.

**registry** = <sqlalchemy.orm.decl_api.registry object>

---

Data descriptors inherited from flask_sqlalchemy.model.Model:

**__dict__**
dictionary for instance variables (if defined)

**__weakref__**
list of weak references to the object (if defined)

---

Data and other attributes inherited from flask_sqlalchemy.model.Model:

**__annotations__** = {'__fsa__': 't.ClassVar[SQLAlchemy]', 'query': 't.ClassVar[Query]', 'query_class': 't.ClassVar[type[Query]]'}

class **MediaMetadataInternalRepresentation**(builtins.object)
MediaMetadataInternalRepresentation(metadataId: int, entryId: int, leftCameraMedia: str, rightCameraMedia: str, time: str, temperature: float, ph: float, dissolvedC

Internal representation for a MediaMetadata entry.

Methods defined here:

**__init__**(self, metadataId: int, entryId: int, leftCameraMedia: str, rightCameraMedia: str, time: str, temperature: float, ph: float, dissolvedOxygen: float)
Construct MediaMetadataInternalRepresentation.

---

Data descriptors defined here:

**__dict__**
dictionary for instance variables (if defined)

**__weakref__**
list of weak references to the object (if defined)

---

Data and other attributes defined here:

**dissolvedOxygen** = 0.0

**entryId** = 0

**leftCameraMedia** = ''

**metadataId** = 0

**ph** = 0.0

**pressure** = 0.0

**rightCameraMedia** = ''

**temperature** = 0.0

**time** = ''

class **illuminationType**(enum.Enum)
illuminationType(value, names=None, *, module=None, qualname=None, type=None, start=1, boundary=None)

Illumination Type.

Method resolution order:
illuminationType
enum.Enum
builtins.object

---

Data and other attributes defined here:

**INFRARED** = <illuminationType.INFRARED: 3>

**NONE** = <illuminationType.NONE: 1>

**ULTRAVIOLET** = <illuminationType.ULTRAVIOLET: 4>

**VISIBLESPECTRUM** = <illuminationType.VISIBLESPECTRUM: 2>

---

Data descriptors inherited from enum.Enum:

**name**
The name of the Enum member.

**value**
The value of the Enum member.

---

Methods inherited from enum.EnumType:

**__contains__**(member) from enum.EnumType
Return True if member is a member of this enum
raises TypeError if member is not an enum member

note: in 3.12 TypeError will no longer be raised, and True will also be
returned if member is the value of a member in this enum

**__getitem__**(name) from enum.EnumType
Return the member matching `name`.

**__iter__**() from enum.EnumType
Return members in definition order.

**__len__**() from enum.EnumType
Return the number of members (no aliases)

---

Readonly properties inherited from enum.EnumType:

**__members__**
Returns a mapping of member name->value.

This mapping lists all enum members, including aliases. Note that this
is a read-only view of the internal mapping.

class **shotType**(enum.Enum)
shotType(value, names=None, *, module=None, qualname=None, type=None, start=1, boundary=None)

Shot Type for the type of study to be performed.

Method resolution order:
shotType
enum.Enum
builtins.object

---

Data and other attributes defined here:

**BURST** = <shotType.BURST: 2>

**SINGLE** = <shotType.SINGLE: 1>

**TELESCOPIC** = <shotType.TELESCOPIC: 3>

**TIMELAPSE** = <shotType.TIMELAPSE: 4>

**VIDEO** = <shotType.VIDEO: 5>

---

Data descriptors inherited from enum.Enum:

**name**
The name of the Enum member.

**value**
The value of the Enum member.

---

Methods inherited from enum.EnumType:

**__contains__**(member) from enum.EnumType

Return True if member is a member of this enum
raises TypeError if member is not an enum member

note: in 3.12 TypeError will no longer be raised, and True will also be returned if member is the value of a member in this enum

**__getitem__**(name) from enum.EnumType
Return the member matching `name`.

**__iter__**() from enum.EnumType
Return members in definition order.

**__len__**() from enum.EnumType
Return the number of members (no aliases)

---

Readonly properties inherited from enum.EnumType:

**__members__**
Returns a mapping of member name->value.

This mapping lists all enum members, including aliases. Note that this is a read-only view of the internal mapping.

## Functions

**getAllMediaEntry**(db) -> list[app.MediaEntryInternalRepresentation]
Get all Media Entries from a database.

**getAllMediaEntryIDs**(db)
Get all entryId from database.

**getAllMediaMetadata**(db) -> list[app.MediaMetadataInternalRepresentation]

**getAllMediaMetadataId**(db, entryId: int) -> list[app.MediaMetadataInternalRepresentation]

**getCurrentTime**() -> str
Return current time in 'yyyy-MM-ddTHH:mm:ss.zzz' format.

**index**()
Return index.html to the / route.

**insertMediaEntry**(db, shotType, illuminationType, iso, apertureSize, shutterSpeed, whiteBalance)
Insert a MediaEntry into the database.

**insertMediaMetadata**(db, entryId, leftCameraMedia, rightCameraMedia, temperature, pressure, ph, dissolvedOxygen)
Insert MediaMetadata entry into database.

## Data

**CONTINUOUS** = <EnumCheck.CONTINUOUS: 'no skipped integer values'>
**UNIQUE** = <EnumCheck.UNIQUE: 'one name per value'>
**app** = <Flask 'app'>
**basedir** = '/home/uwu/Github_Repos/University/M.O.S.I.S/M.O.S.I.S_Host_Software/App'
**db** = <SQLAlchemy>